

# The Students Guide to Kalah and the GameLink Platform

## Rules of Kalah:

- Kalah is a 2 player turn based game. At the start of the game, 3 seeds are placed in each house
- Each player controls the six houses and their seeds on the player's side of the board. The player's score is the number of seeds in the store to their right
- Players take turns planting their seeds. On a turn, the player removes all seeds from one of the houses under their control. Moving counter-clockwise, the player drops one seed in each
- If the last planted seed lands in the player's store, the player gets an additional move. There is no limit on the number of moves a player can make in their turn
- If the last planted seed lands in an empty house owned by the player, and the opposite house contains seeds, both the last placed seed and the opposite seeds are captured and placed into the player's store
- When one player no longer has any seeds in any of their houses, the game ends. All remaining seeds on the other side are moved into the other player's store, and the player with the most seeds in their store wins

**Note:** To see the game in action go to <http://www.worldtreesoftware.com/apps/kalah/kalah.html>

## Score:

The score of the game is determined by the number of seeds in your store, the person with the most seeds in their store at the end of the game wins. If a player enters an invalid move: the player will immediately lose the match and all remaining seeds on the board will be placed in the winning player's store.

Game board at the start of the game:

0	3	3	3	3	3	3	0
	3	3	3	3	3	3	
Player 1's turn							

Player 1 makes move: 4 and gets to move again

0	3	3	3	3	3	3	1
	3	3	3	0	4	4	
Player 1's turn							

Player 1 makes move: 1 and steals 3 seeds

0	3	3	3	0	3	3	5
	0	4	4	0	4	4	
Player 2's turn							

Players can no longer move: game is over

17	0	0	0	0	0	0	9
	0	0	2	0	0	8	

Game board is wiped, player 1 wins

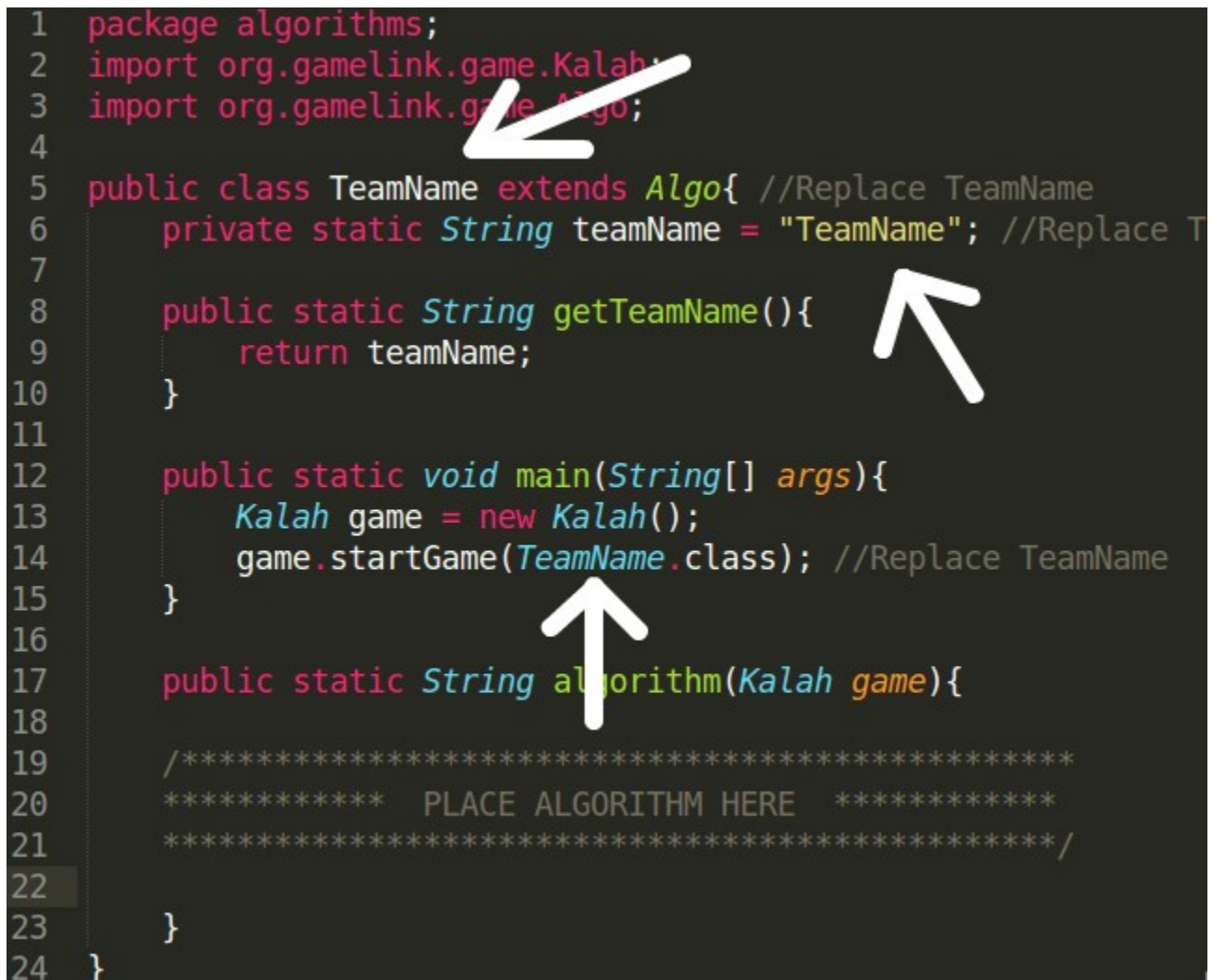
17	0	0	0	0	0	0	19
	0	0	0	0	0	0	
Gameover: You win!							

**Final Score:** Player 1: 19, Player 2: 17

# Setting up the GameLink Platform

The gameLink platform is a java program that allows the player's algorithm to play the Kalah game with other players on a LAN network. Please follow these steps to set up your algorithm with the GameLink platform

1. Download the Kalah zip from <https://github.com/mysteryhobo/Kalah> and extract it to the directory of your choosing.
2. Navigate to the algorithms folder and open the TeamName.java file
  - Note: This java source file is where you will be doing all your work for the project
3. Apply the following changes to the java source file
  - On all lines marked "Replace TeamName" (3 total), replace "TeamName" with your actual team name (Refer to the image below for locations). Your team name must be unique, and must adhere to the restrictions of a java class name.
  - Insert your algorithm into the algorithm method (refer to page 4 for more details)
  - Save this new java file with the name of your team followed by .java (Example: Batman.java if your team name was batman)



```
1 package algorithms;
2 import org.gamelink.game.Kalah;
3 import org.gamelink.game.Algo;
4
5 public class TeamName extends Algo{ //Replace TeamName
6     private static String teamName = "TeamName"; //Replace T
7
8     public static String getTeamName(){
9         return teamName;
10    }
11
12    public static void main(String[] args){
13        Kalah game = new Kalah();
14        game.startGame(TeamName.class); //Replace TeamName
15    }
16
17    public static String algorithm(Kalah game){
18
19        /*****
20        ***** PLACE ALGORITHM HERE *****
21        *****/
22
23    }
24 }
```

*Locations of "TeamName"s to be changed*

## Compiling and Running the Game:

The following steps show how to compile and run the Kalah game using the command line, if you wish to use an IDE, please refer to your IDE's manual for information on importing a project.

### Option 1: Compiling and Running Using Batch or Makefile (Recommended):

#### Windows (Batch file):

1. Open the Kalah-master folder and open the batch.txt file in a text editor (notepad, sublime, etc..)
2. replace "TEAM\_NAME\_HERE" on the first line with the team name you used in your algorithm (must be exactly the same)
3. Save the text file to the Kalah-master folder with the name of your team followed by the .bat extension. Example: TeamName.bat
4. Double click on the batch file to run the program.

#### Linux (makefile):

1. Open the file titled make file in a text editor
2. On the first line replace "TEAM\_NAME\_HERE" with the team name used in your algorithm (must be exactly the same)
3. Save the edited makefile with the name makefile (no extension)
4. Change the directory of the terminal to the Kalah-master folder using the cd command
5. run the program by entering "make" into the terminal

### Option 2: Compiling and Running Using Command Line:

1. Open the command Line (cmd for windows, terminal for linux).
2. Using the change directory command (cd <filename>) change the command line directory to the Kalah-master folder that you downloaded.
3. Once in the Kalah-master directory you can enter the following command to compile your algorithm. Remember to replace TeamName.java with your actual algorithm java file name that you made in step 3 of the previous page.

```
javac -cp Kalah.jar algorithms/TeamName.java
```

4. You may now run your algorithm with the game by entering the following command. Remember to replace TeamName with your actual algorithm java file name

```
Windows: java -cp Kalah.jar;. algorithms.TeamName
Linux: java -cp Kalah.jar;. algorithms.TeamName
```

If successful you should see something like the image below:

```
*****
CONFIG: Inital Cells Filled = 2
CONFIG: Board Height = 5
CONFIG: Board Width = 5
CONFIG: Time Limit = 30
CONFIG: Time Delay = 3
CONFIG: Display Board = true
*****
Please enter your player number <1 or 2>:
```

## Connecting:

The Gamelink platform connects players via a peer to peer connection over a local area network(LAN). Follow the instructions outline how to connect as each player:

**Connecting as Player1:** Run the game, enter “1” to identify as player 1, and wait for player2 to connect (Player1's IP address is then displayed in the command prompt).

**Connecting as Player2:** Run the game and enter “2” to identify as player 2 then enter the IP address of Player1

## Coding your algorithm

Your algorithm will return a String containing 1 number, this number represents which house you wish to move; houses will be number 1 – 6. Where the number 1 represents the leftmost house, and the number 6 represents the rightmost house. Your algorithm will be called each time it is your turn, meaning do not return a sting of multiple integers when you have multiple turns in a row.

### Algorithm Rules:

- Any String returned by the algorithm not following the standard outlined above will result in an invalid move and an immediate loss of the match.
- Attempting to make a move on an empty house will also result in an invalid move and an immediate loss of the match.
- Any algorithm that exceeds the time specified by the move time limit setting will result in an invalid move and an immediate loss of the match.
- Any algorithm that encounters an error while running will result in an invalid move and an immediate loss of the match.

### game.getBoard();

Your algorithm can call the game.getBoard() method to return the game board in the form of a 2d integer array. Where the integers represent the number of seeds in that particular house / store. Please refer to the table below to see the layout of the 2d array returned by the game.getBoard() method.

Opponent's Side							
Opponent's Store	House 6	house 5	house 4	house 3	house 2	house 1	Buffer
Buffer	House 1	House 2	House 3	House 4	House 5	House 6	Your Store
Your Side							

The table below details the positions of the cells within the java 2d array.

Your Houses:	Your Store:	Opponent's Houses	Opponent's Store
Row: 1 column: 1 - 6	Row: 1 Column: 7	Row: 0 Column: 1 - 6	Row: 0 Column: 0

The method should be used as follows:     `int[][] gameBoard = game.getBoard();`

**Note:** Do not alter any of the pre-written code as it may cause errors when marking your algorithm. You can however, add your own imports and methods to be called from within you algorithm method.

## Game Settings:

While testing your algorithm you may want to change some of the game settings. This can be done by changing the values in the `Kalah.properties` file in the `classes` folder. The following is a list of all the properties and what they do:

- **Seeds:** The `Seeds` Property dictates the number of seeds in each house at the beginning of the game
- **Houses:** The `houses` property dictates the number house each player will have on their side
- **MoveTimeLimit:** The `MoveTimeLimit` property dictates the maximum amount of time your algorithm can take to make a move (in seconds).
- **DisplayBoard:** The `displayBoard` property dictates whether or not the board will be displayed after each move.
- **MoveDelay:** The `MoveDelay` property adds a delay (in seconds) between moves therefore slowing the game, allowing users to follow the playing of the game

## Playing Against yourself:

If you wish to play your algorithm against itself, or play different algorithms against each other for comparison, you can do so using the following steps:

### Against itself:

To play the algorithm against itself simple run the algorithm in two different command lines, one as player 1 and player 2 and follow the instructions outlined in the `Connecting` section

### Creating another algorithm to play against:

1. Create a copy of your current algorithm java source file.
2. Repeat the changes conducted in step 3 from `Setting up the Gamelink platform` with a different team name (team name must be different than the one used previously, but must adhere to the same java class naming restrictions).
3. Compile and run both algorithms in separate command lines, one as `player1` the other as `player2`

## The Tournament and submitting your algorithm:

A round-robin style tournament will be conducted to determine the strength of the algorithms. Each algorithm will face off against all others twice; once as player 1 and once as player 2 to ensure fairness. Algorithms will be ranked in order of points earned, where winning a match awards 2 points and tying awards 1. In the event of a tie in points, the sum of all game scores will act as the tie breaker.

**Submission:** To submit your algorithm simply upload your `.java` file to blackboard, be sure to only upload the `.java` file and not the whole `Kalah-master` folder.

**NOTE:** Do not alter any of the pre-written code. If you submit a `.java` file with altered pre-written code (other than that specified in the `Setting up the GameLink platform` section) your algorithm will not be runnable in the tournament, resulting in a loss of every match and a total score of zero.

If you have any questions you can email [peter.little1@uoit.net](mailto:peter.little1@uoit.net)